CONCURRENT ALGORITHMS FOR TRANSIENT FE ANALYSIS

by

M. Ortiz

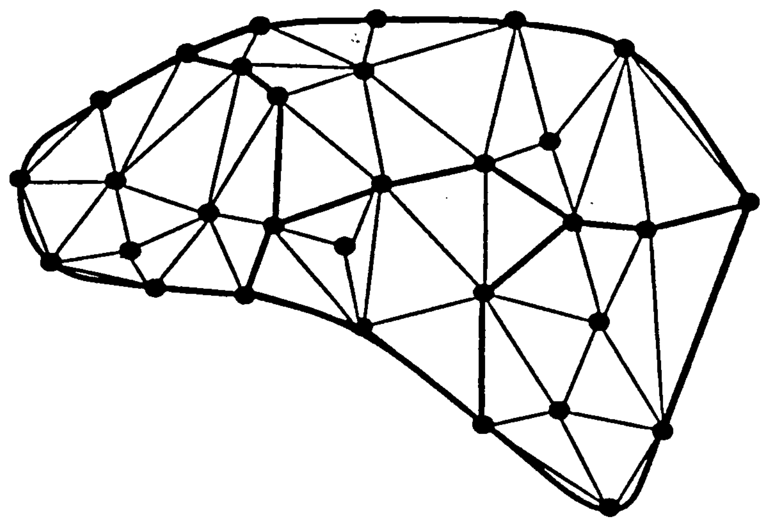
Brown University
Division of Engineering
Providence, RI 02912

and

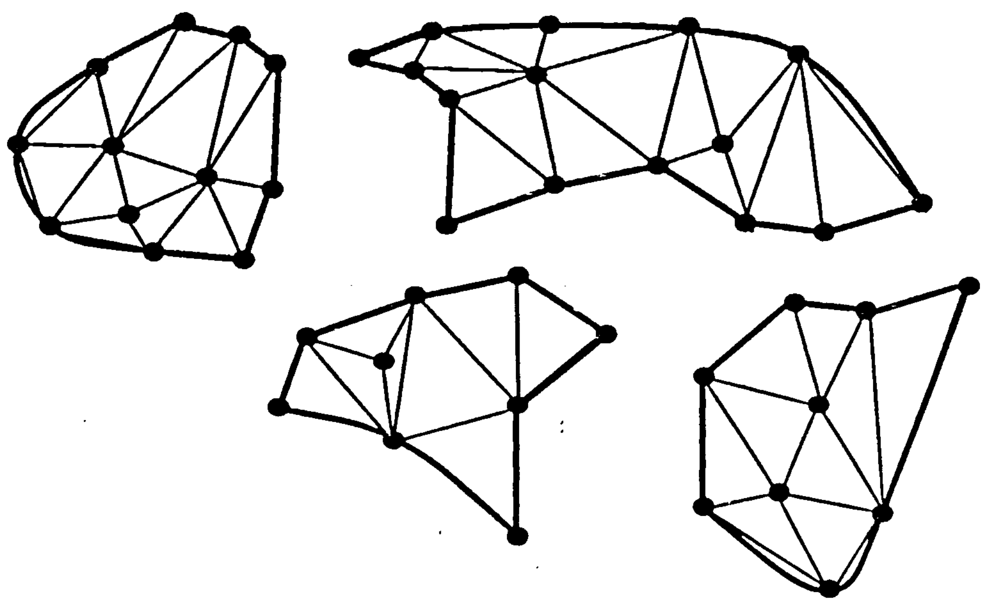
B. Nour-Omid

Mechanics and Materials Lab.
Lockheed Palo Alto Research Lab.
Palo Alto, CA 94304

'CUT AND PASTE' ALGORITHMS



MODEL STRUCTURE



THE STRUCTURE VIEWED AS A COLLECTION OF DISCONNECTED SUBSTRUCTURES

THIS PAGE LEFT BLANK INTENTIONALLY

A 'CUT AND PASTE' ALGORITHM

- Predictor phase:

$$\tilde{\mathbf{d}}_{n+1} = \mathbf{d}_n + \triangle t\mathbf{v}_n + (1/2 - \beta)\triangle t^2\mathbf{a}_n$$
$$\tilde{\mathbf{v}}_{n+1} = \mathbf{v}_n + (1-\gamma)\triangle t\mathbf{a}_n$$

- Equation solving phase:

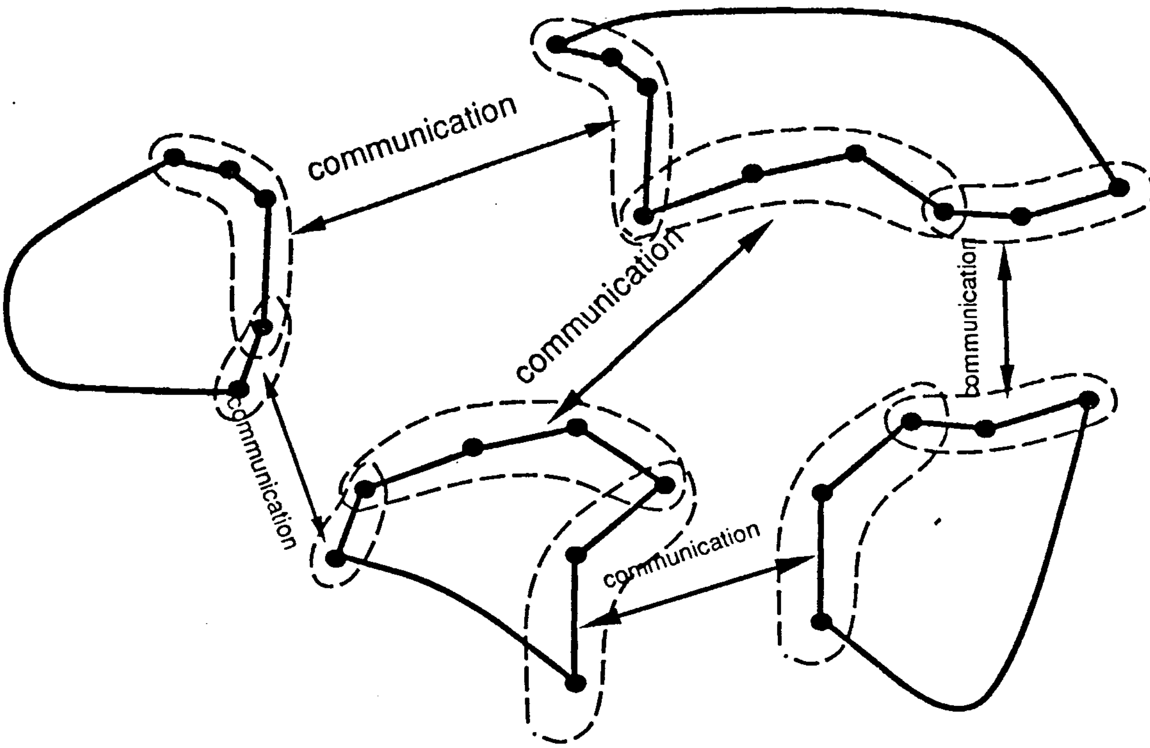
$$\mathbf{a}_{n+1} = \mathbf{0}$$
$$for\ s = 1, NS\ do$$
$$\tilde{\mathbf{a}}^s_{n+1} = -(\mathbf{M}^s + \beta\triangle t^2\mathbf{K}^s)^{-1}\mathbf{K}^s\tilde{\mathbf{d}}^s_{n+1}$$
$$\mathbf{a}_{n+1} = \mathbf{a}_{n+1} + \mathbf{M}^s\tilde{\mathbf{a}}^s_{n+1}$$
$$\mathbf{a}_{n+1} = \mathbf{M}^{-1}\mathbf{a}_{n+1}$$

- Corrector phase:

$$\mathbf{d}_{n+1} = \tilde{\mathbf{d}}_{n+1} + \beta\triangle t^2\mathbf{a}_{n+1}$$
$$\mathbf{v}_{n+1} = \tilde{\mathbf{v}}_{n+1} + \gamma\triangle t\mathbf{a}_{n+1}$$

Definition of concurrent dynamic algorithms. Note three-phase structure similar to that of Newmark's method. Equation solving phase involves subdomain factorizations only. Local accelerations are mass-averaged at the subdomain boundaries.

INTERPROCESSOR COMMUNICATIONS



REDUCED SUBSTRUCTURES SHOWING THE COMMUNICATION DUE TO SHARED DEGREES OF FREEDOM .

THIS PAGE LEFT BLANK INTENTIONALLY

OVERVIEW OF GENERAL PROPERTIES

- Parameters:

  $n =$ Number of dof in structure.
  $s =$ Number of element groups.
  $p =$ Number of processors.
  $i =$ Number of interface dof.

- General properties:

  i) Newmark's method is obtained for $s = 1$.

  ii) Unconditional stability for all s and $\gamma \geq 1/2$, $\beta \geq \gamma/2$.

  iii) Full concurrency on a p-processor machine ($p \leq n$) except for $O(i)$ operation (mass-averaging).

  iv) For given accuracy and $n/s \to \infty$,

$$SPEED - UP = \begin{cases} O(p\sqrt{s}), & (2D) \\ O(ps), & (3D) \end{cases}$$

PRECEDING PAGE BLANK NOT FILMED

PAGE 1073 INTENTIONALLY BLANK

General properties. Note two-parameter dependence of speed-up estimates on both the number of processors and the number of subdomains.

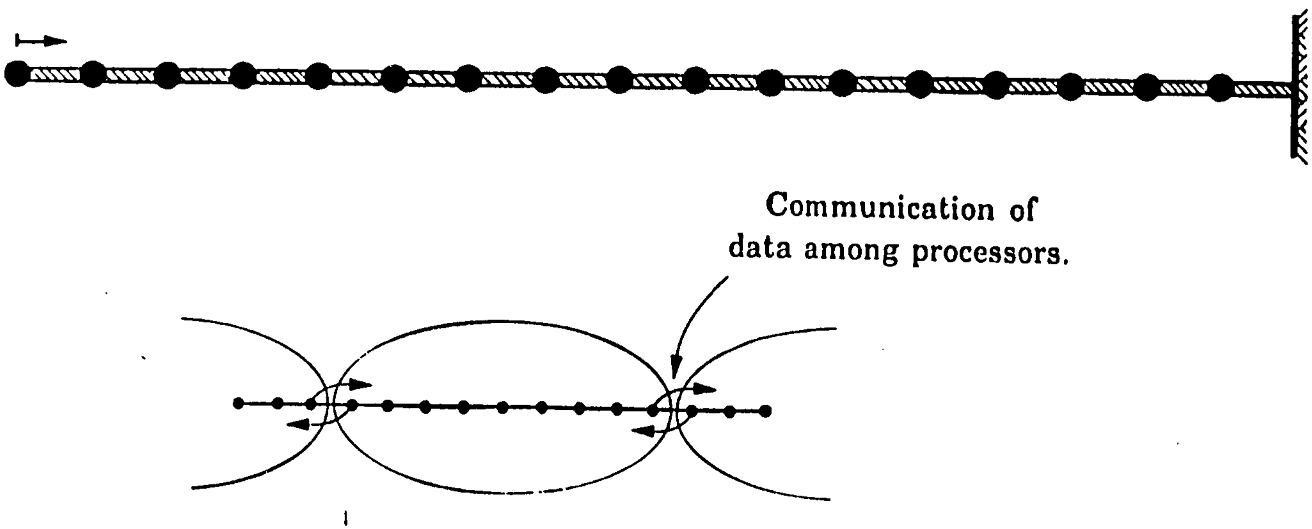


Figure 3. Discretization and partition of the bar problem.

One-dimensional test for assessing communication efficiencies.

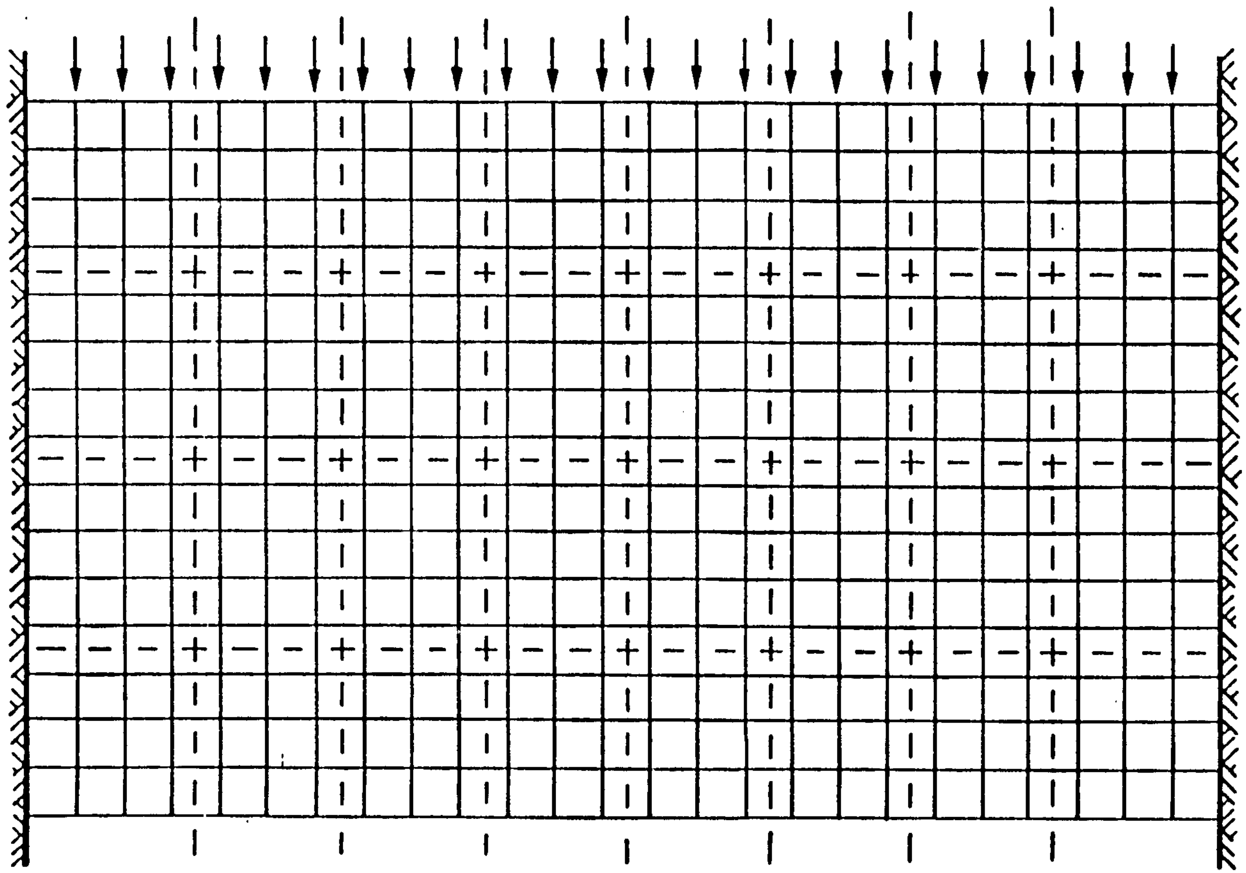


Figure 4. Discretization and partition of the plane stress problem on a 32-processor computer.

Two-dimensional test for assessing communication efficiencies.

N	Computation time (milisec.)	Communication time (milisec.)	efficiency %	Rate Mflop
64	1.47	0.46	76.32	0.233
96	2.20	0.46	82.86	0.253
160	3.67	0.46	88.96	0.271
288	6.61	0.46	93.55	0.285
544	12.49	0.46	96.48	0.294

Table 5. Performance of the Bar problem on the 32 Processor Hypercube.

No. of Elements	Computation time (milisec.)	Communication time (milisec.)	efficiency %	Rate Mflop
128	25.2	7.8	76.3	0.232
288	57.1	8.6	87.0	0.265
800	161.3	10.6	93.9	0.286
2592	530.9	14.6	97.3	0.297
9248	1915.2	22.6	98.8	0.301

Communication efficiencies for 1-D and 2-D test cases. Note efficiencies approaching 99.

COMPUTATIONAL EFFICIENCY

- $COST \approx \frac{1}{2}nb^2 + 2nb$, ($b =$ semi-bandwidth)
- Square mesh, l^2 elements:

$$GLOBAL \approx \frac{1}{2}(l+2)^2(l+1)^2 + 2(l+2)(l+1)^2$$

- Partitioned mesh, $s = m^2$ subdomains:

$$PARTITIONED \approx$$

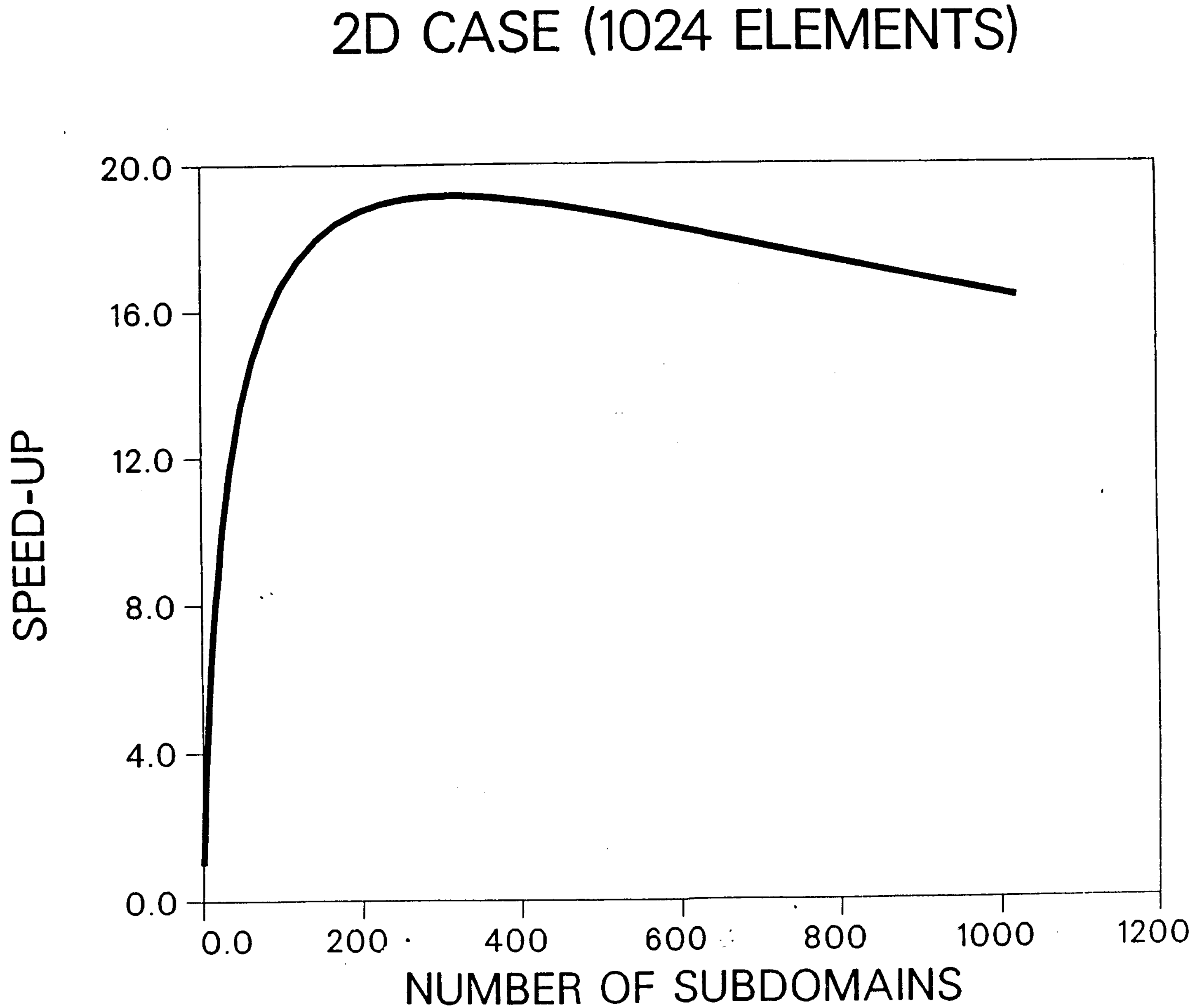
$$s\left[\frac{1}{2}\left(\frac{l}{m}+2\right)^2\left(\frac{l}{m}+1\right)^2 + 2\left(\frac{l}{m}+2\right)\left(\frac{l}{m}+1\right)^2\right]$$

- Equation solving speed-up ($n/s \to \infty$):

$$SPEED-UP(2D) = \frac{GLOBAL}{PARTITIONED} \approx O(s)$$

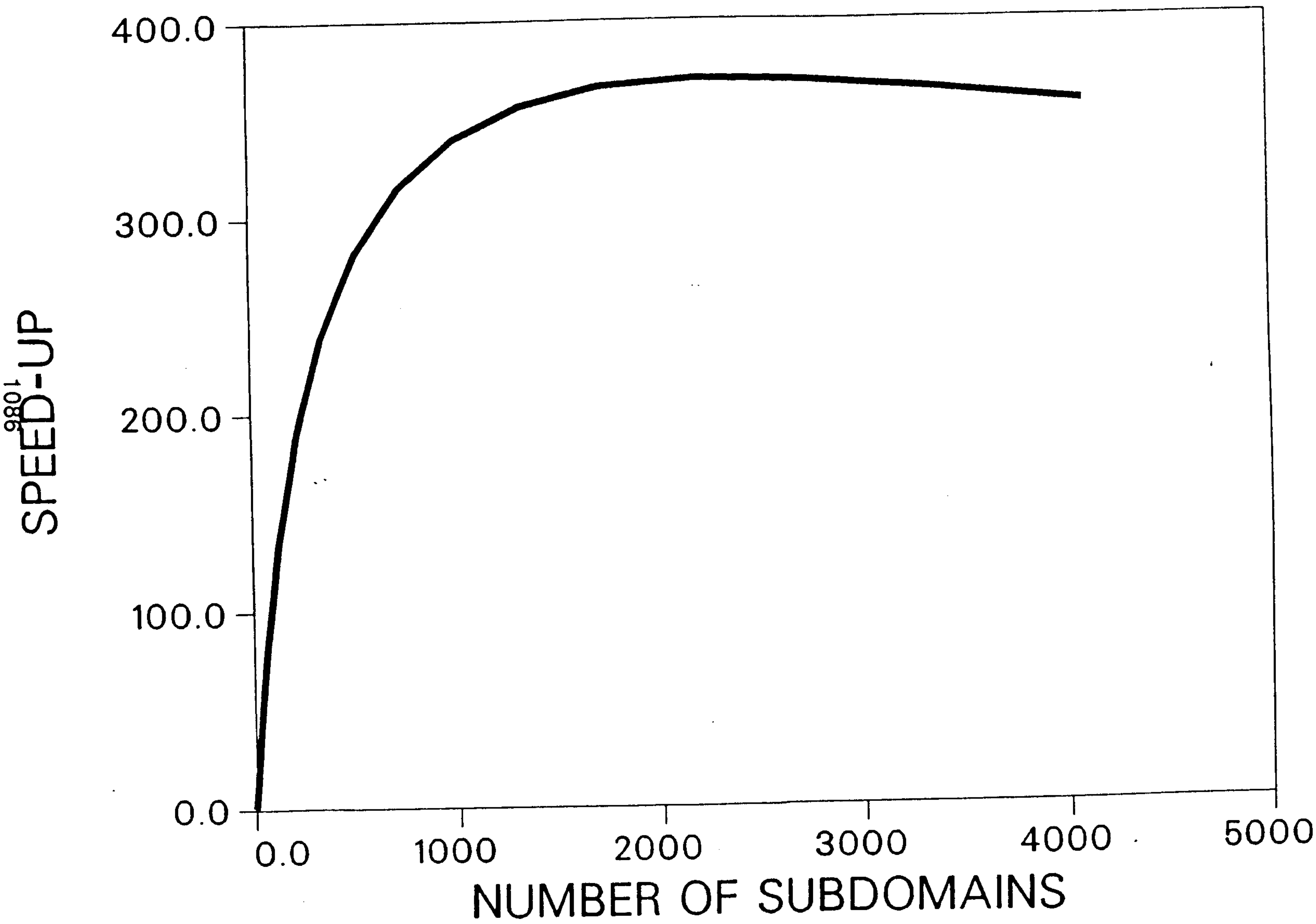
$$SPEED-UP(3D) = \frac{GLOBAL}{PARTITIONED} \approx O(s^{4/3})$$

Estimated equation solution speed-up for one application of the algorithm. Savings arise from the smaller dimensionality of the matrices being factorized and from the fact that the interface nodes are never assembled.


2D CASE (1024 ELEMENTS)
SPEED-UP
20.0
16.0
12.0
8.0
4.0
0.0
0.0
200
400
600
800
1000
1200
NUMBER OF SUBDOMAINS

Estimated equation solution speed-up for one application of the algorithm vs. number of subdomains, 2-D case.

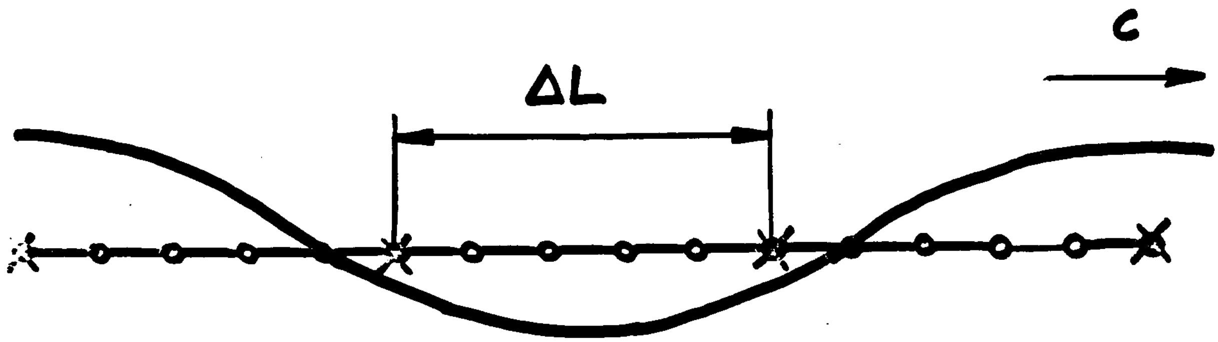
3D CASE (4096 ELEMENTS)



Estimated equation solution speed-up for one application of the algorithm vs. number of subdomains, 3-D case.

ACCURACY ANALYSIS

- **Algorithmic phase errors, 1D case:**



- **Maximum celerity of computed waves**

$$c_{max} = \Delta L / \Delta t$$

$\Delta L \equiv$ **subdomain size**

- **For accurate results, need to take**

$$c_{max} \geq c, \qquad OR \quad \Delta t \leq \Delta L / c$$

$c \equiv$ **wave celerity**

Accuracy requirements derived from an analysis of phase errors in one dimension. Note the Courant-type condition on the time step to preserve the level of accuracy as the size of the subdomains is increased.

ACCURACY REQUIREMENTS

- Square mesh, $s = m^2$ subdomains:

$$\Delta L = L/m \approx O(1/\sqrt{s})$$
$$\Delta t \leq \Delta L/c = L/mc \approx O(1/\sqrt{s})$$

- Net speed-up ($p = 1$):

$$SPEED-UP(2D) \approx O(s) \times O(1/\sqrt{s}) = O(\sqrt{s})$$

- Cubic mesh, $s = m^3$ subdomains:

$$\Delta L = L/m \approx O(1/s^{1/3})$$
$$\Delta t \leq \Delta L/c = L/mc \approx O(1/s^{1/3})$$

- Net speed-up ($p = 1$):

$$SPEED-UP(3D) \approx O(s^{4/3}) \times O(1/s^{1/3}) = O(s)$$

Estimated equation solution speed-ups for a square mesh in large scale nonlinear computations, for a prescribed level of accuracy. The net speed-up is computed by taking into account both the savings in equation solving afforded by the method as well as the reduction in time step necessary to keep the level of accuracy unchanged as the number of subdomains is increased.

NUMERICAL TESTS

- Square membrane, simply supported, subjected to uniform initial velocity.

- Finite deflection FE formulation. Triangular elements:

$$W = \frac{T}{2}\frac{A^2}{A_0}$$

T = tension
A_0 = initial area of triangle
A = deformed area.

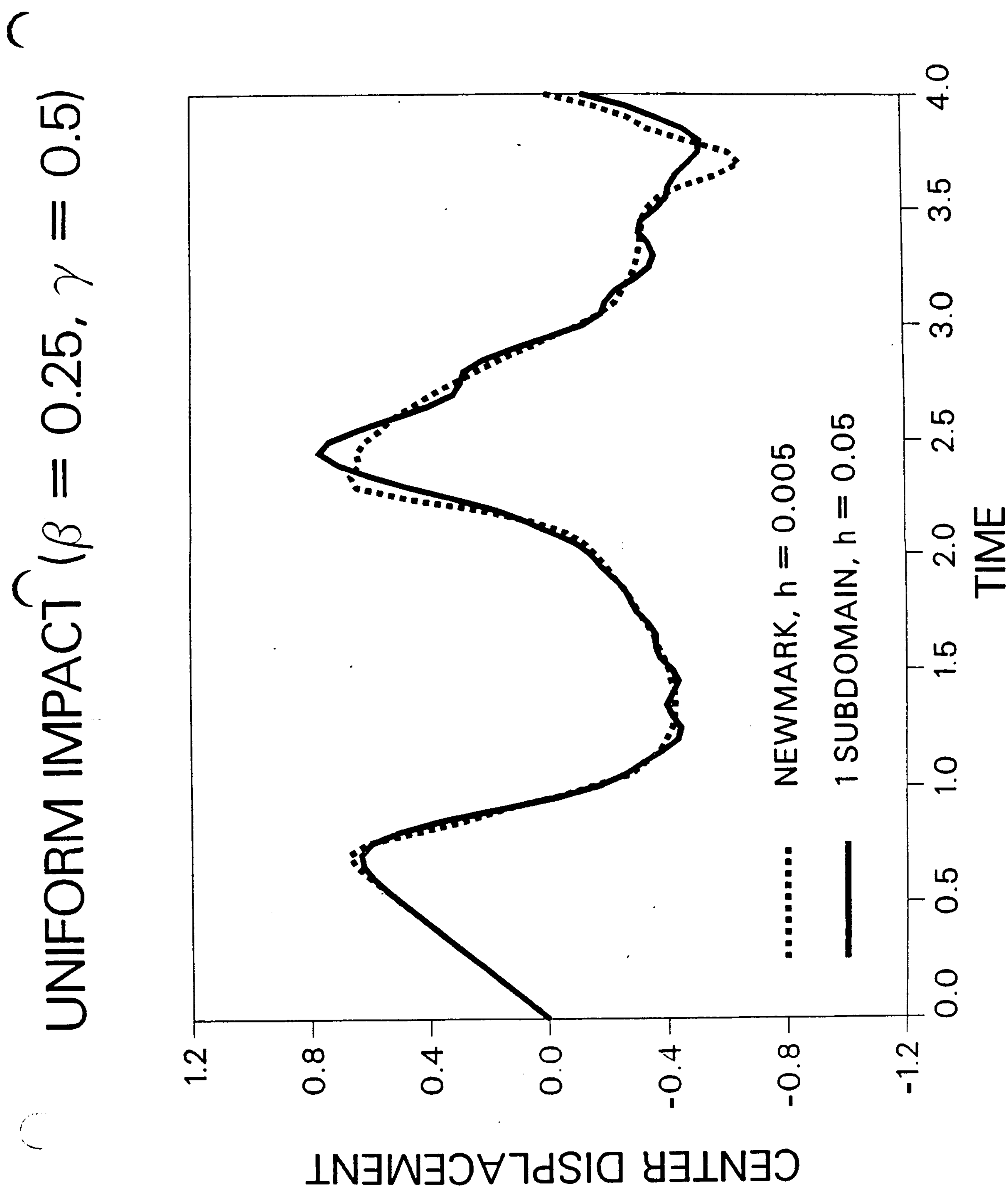
- Quadrilateral elements:

- Parameters: $L = 2$, $T = 1$, $\rho = 1$.

- Error measure:

$$ERROR = \left[\int_0^T \mid w(t) - w_{exact}(t) \mid^2 \frac{dt}{t^2} \right]^{1/2}$$

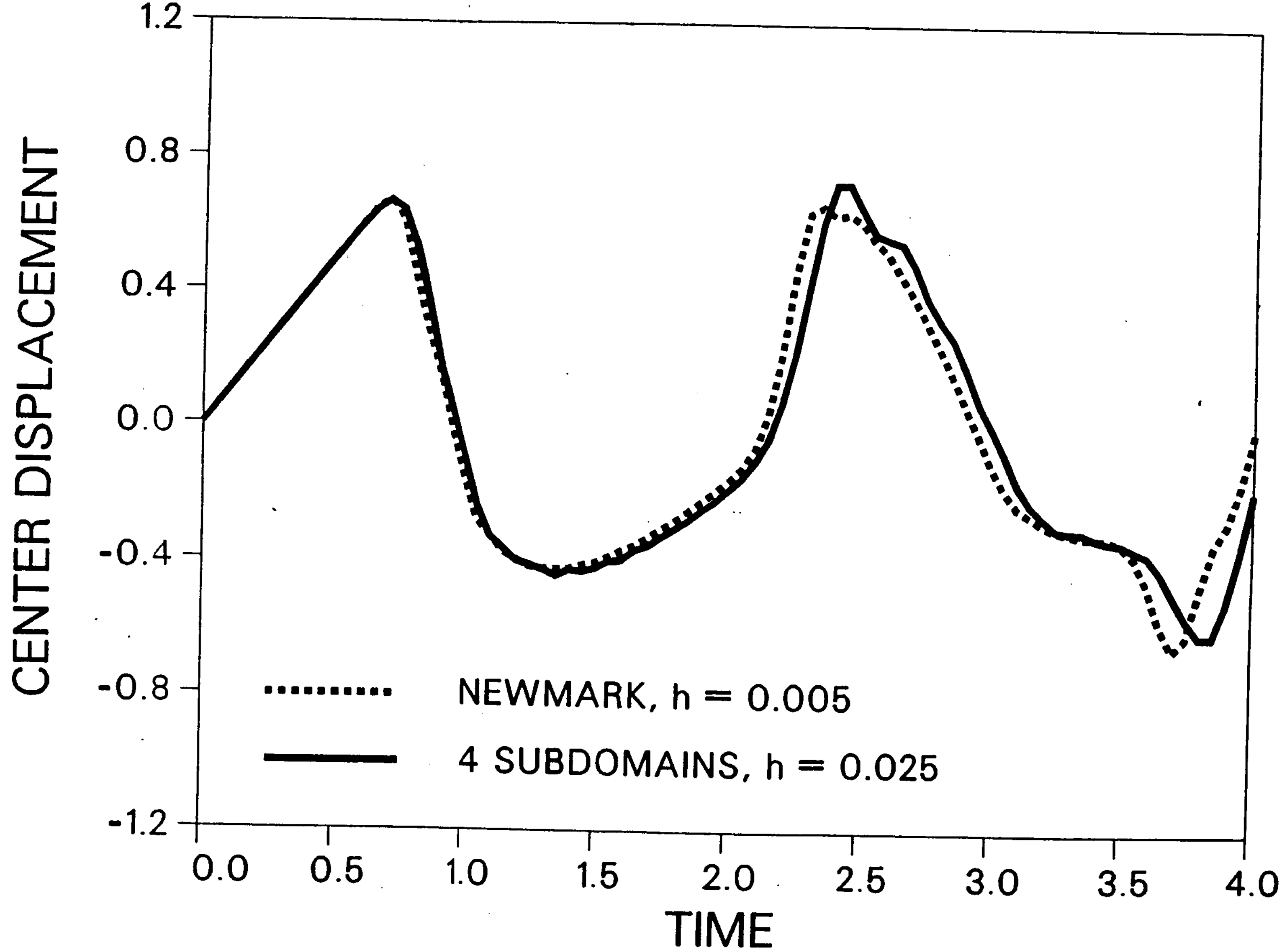
THIS PAGE LEFT BLANK INTENTIONALLY



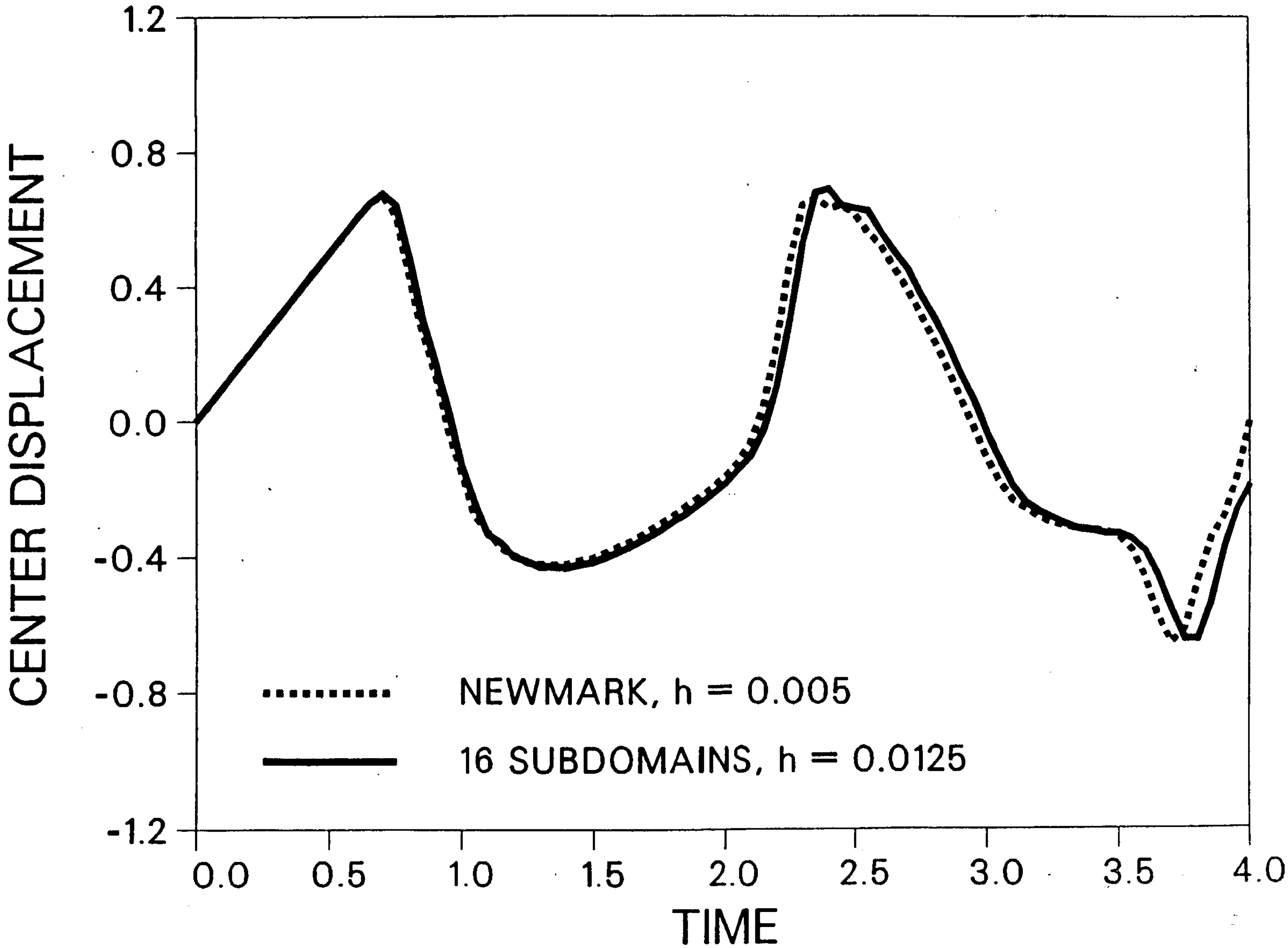
PAGE 1093 INTENTIONALLY BLANK

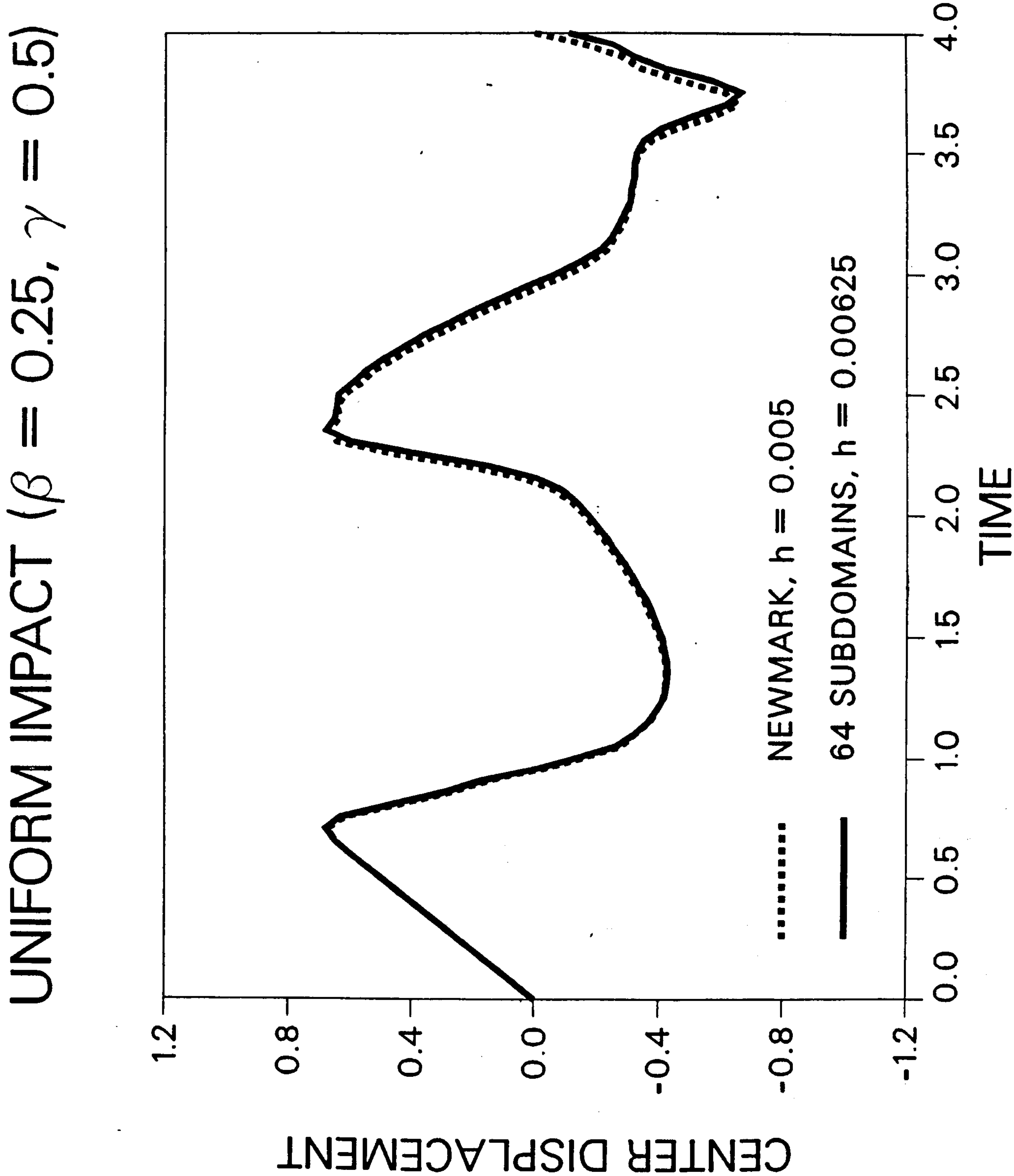
PRECEDING PAGE BLANK NOT FILMED

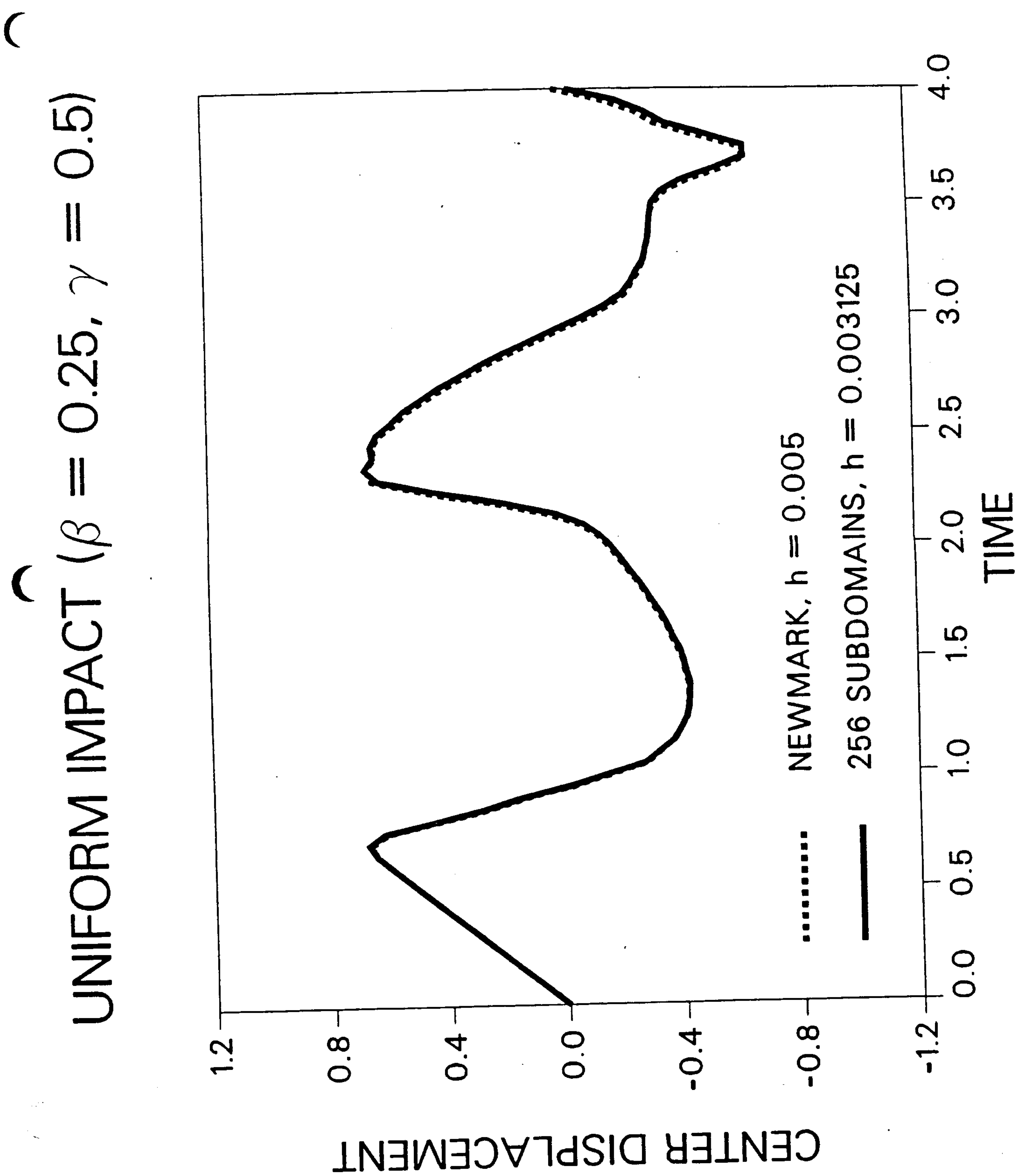
UNIFORM IMPACT ($\beta = 0.25$, $\gamma = 0.5$)



UNIFORM IMPACT ($\beta = 0.25$, $\gamma = 0.5$)

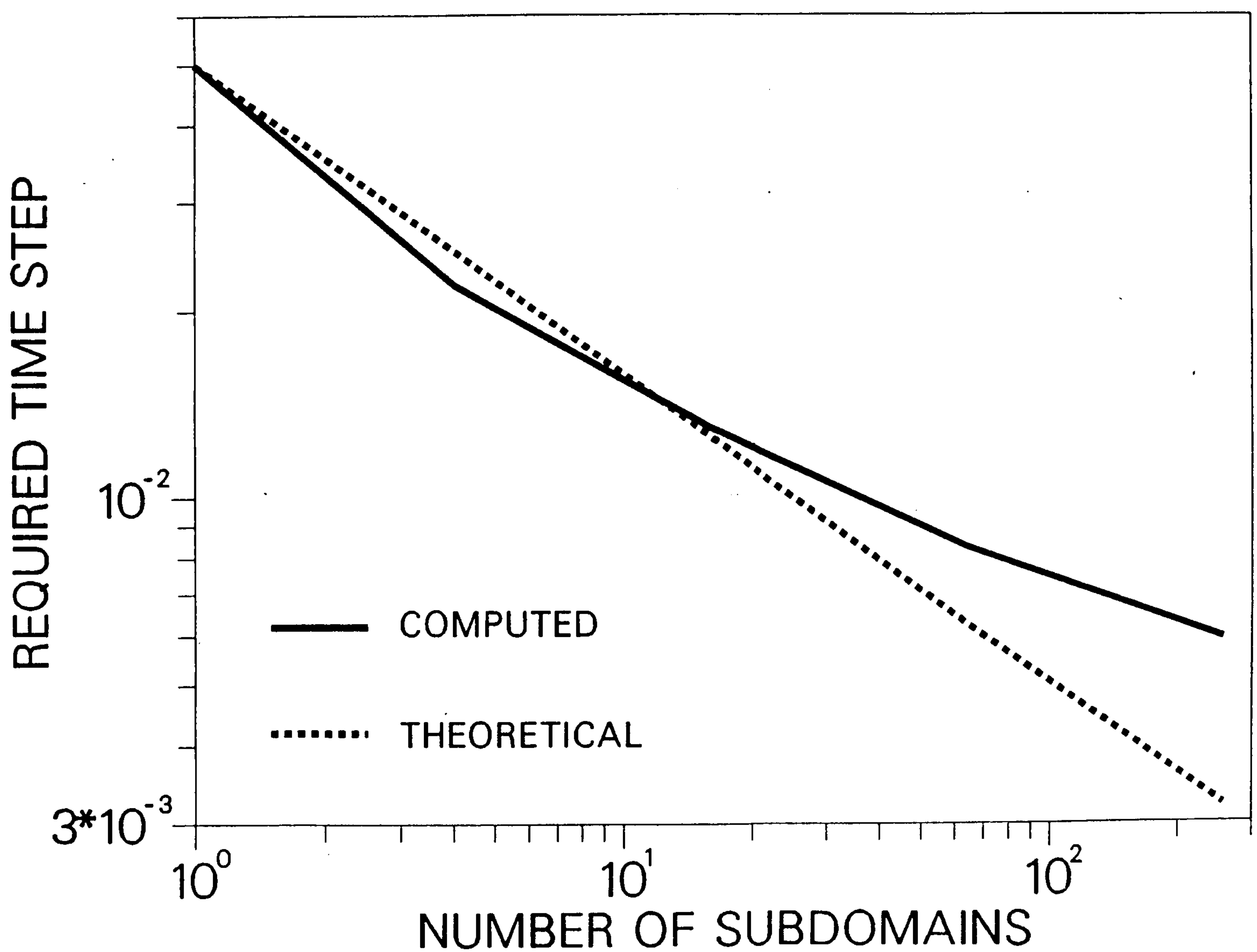



UNIFORM IMPACT (β = 0.25, γ = 0.5)
CENTER DISPLACEMENT
TIME
1.2
0.8
0.4
0.0
-0.4
-0.8
-1.2
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
NEWMARK, h = 0.005
64 SUBDOMAINS, h = 0.00625


UNIFORM IMPACT (β = 0.25, γ = 0.5)
CENTER DISPLACEMENT
1.2
0.8
0.4
0.0
-0.4
-0.8
-1.2
TIME
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
NEWMARK, h = 0.005
256 SUBDOMAINS, h = 0.003125

Numerical test designed to measure directly the actual speed-ups associated with the method, for a prescribed level of accuracy in the solution. Problem concerns square membrane undergoing finite deflections. Note the definition of the error at the center of the membrane used to monitor accuracy.

ACCURACY REQUIREMENTS



Actual vs. estimated time step requirements as a function of number of subdomains. Theoretical time steps follow from the Courant-type accuracy condition. Actual requirements are determined by monitoring accuracy directly in square membrane test.

SINGLE-PROCESSOR SPEED-UPS

1024 ELEMENT CASE			
NSUB	Secs.	Speed-up	Theory
1	1143	1	1
4	776	1.47	2
16	521	2.19	4
64	326	3.51	8
256	156	7.31	16

Actual vs. estimated speed-ups for square membrane problem on a single processor. Timings correspond to the equation-solving phase only. Time steps were chosen so as to obtain the same level of accuracy from all runs.